

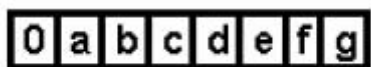
Lista de Exercícios 2

Conceitos envolvidos, além dos desenvolvidos na Lista 1:

- a) Interface com Teclado Matricial
- b) Interface com Displays de 7 segmentos
- c) Interface com LCD
- d) Interface com Conversores A/D e D/A
- e) Interface usando Mapeamento de Memória

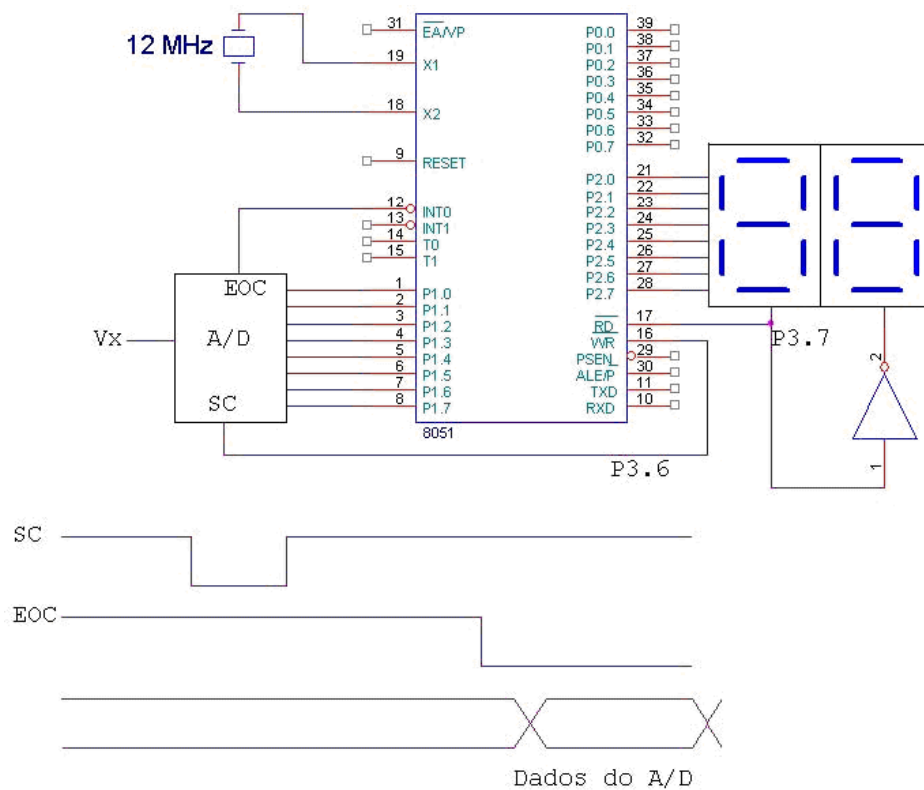
1) Para o esquema de um Voltímetro com fundo de escala de 15 volts e com resolução de 1 volt construído com o Microcontrolador 8051, considerar os seguintes dados :

- O Conversor A/D tem tempo de conversão de 100 us (microssegundos) isto é, este é o tempo decorrido entre a subida de borda do pulso SC (Start Conversion) a ser gerado pelo microcontrolador no pino P3.6 e a descida de borda gerada pelo A/D através do sinal EOC (End of Conversion) indicando que os dados estão disponíveis para serem lidos.
- A duração do pulso de SC (Start Conversion) deve ser de no mínimo 50 us (microssegundos).
- Os dois Displays de 7 segmentos são do tipo catodo comum e multiplexados através do pino P3.7 que comanda qual Display será aceso no tempo.
- O valor a ser aceso no Display deve ser disponibilizado na Porta P2 na ordem:



O Pino P2.7 não está sendo usado e cada Display mostrará valores de 0 a 9.

Escrever o Programa Principal que deve converter os dados armazenados no Buffer de Display (endereços da RAM Interna 30h = LSB e 31h = MSB) para 7 segmentos e enviá-los para os Displays Multiplexados, permanecendo continuamente em loop. Programar a Interrupção INT0 para responder ao Conversor A/D. Os valores armazenados no Buffer de Display vão de 00 a 09 para o endereço 30h e de 00 a 01 para o endereço 31h .



Escrever a sub-rotina de atendimento de Interrupção do Conversor A/D que deverá ler o valor Digital equivalente da Tensão Vx, de acordo com o diagrama de tempos fornecido, e armazená-lo nas posições 30h e 31h da RAM interna (Buffer dos Displays) da seguinte forma:

Valor em P1 – Displays – Buffer dos Displays

De 00 a 0F → 00 volts → 31h = 00 e 30h = 00
 De 10 a 1F → 01 volts → 31h = 00 e 30h = 01
 De 20 a 2F → 02 volts → 31h = 00 e 30h = 02
 De 30 a 3F → 03 volts → 31h = 00 e 30h = 03
 De 40 a 4F → 04 volts → 31h = 00 e 30h = 04
 De 50 a 5F → 05 volts → 31h = 00 e 30h = 05
 De 60 a 6F → 06 volts → 31h = 00 e 30h = 06
 De 70 a 7F → 07 volts → 31h = 00 e 30h = 07
 De 80 a 8F → 08 volts → 31h = 00 e 30h = 08
 De 90 a 9F → 09 volts → 31h = 00 e 30h = 09
 De A0 a AF → 10 volts → 31h = 01 e 30h = 00
 De B0 a BF → 11 volts → 31h = 01 e 30h = 01
 De C0 a CF → 12 volts → 31h = 01 e 30h = 02
 De D0 a DF → 12 volts → 31h = 01 e 30h = 03
 De E0 a EF → 14 volts → 31h = 01 e 30h = 04
 De F0 a FF → 15 volts → 31h = 01 e 30h = 05

0v --- 00
1v --- 17h
2v --- 2Eh
3v --- 45h
4v --- 5Ch
5v --- 73h
6v --- 8Ah
7v --- A1h
8v --- B8h
9v --- CFh
10v --- E6h

VALOR ANALOGICO DA TENSÃO NO SENSOR y = xx volts

onde xx é um número de 00 a 10 equivalente à tensão na entrada INy e y é um número de 0 a 7 equivalente ao sensor ligado a ela. O programa deve enviar oito valores acompanhados de suas respectivas informações como explicado acima, e ficar em loop para que quando ocorra uma mudança nos sensores, esta mudança apareça diretamente no Monitor de Vídeo.

O Clock do conversor deve ser gerado pelo Timer 0 e deve ser de aproximadamente 50 KHz.

Exemplo da tela do Monitor de Vídeo:

VALOR ANALOGICO DA TENSAO NO SENSOR 0 = 05 volts

VALOR ANALOGICO DA TENSAO NO SENSOR 1 = 09 volts

VALOR ANALOGICO DA TENSAO NO SENSOR 2 = 00 volts

VALOR ANALOGICO DA TENSAO NO SENSOR 3 = 03 volts

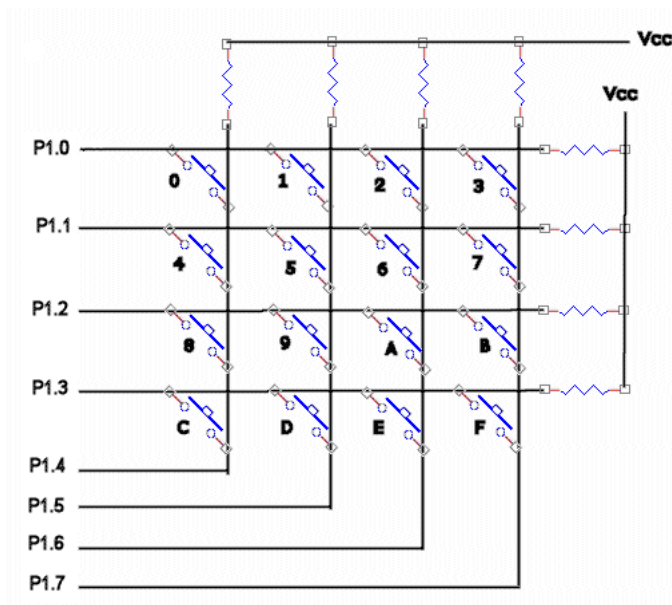
VALOR ANALOGICO DA TENSAO NO SENSOR 4 = 10 volts

VALOR ANALOGICO DA TENSAO NO SENSOR 5 = 01 volts

VALOR ANALOGICO DA TENSAO NO SENSOR 6 = 04 volts

VALOR ANALOGICO DA TENSAO NO SENSOR 7 = 08 volts

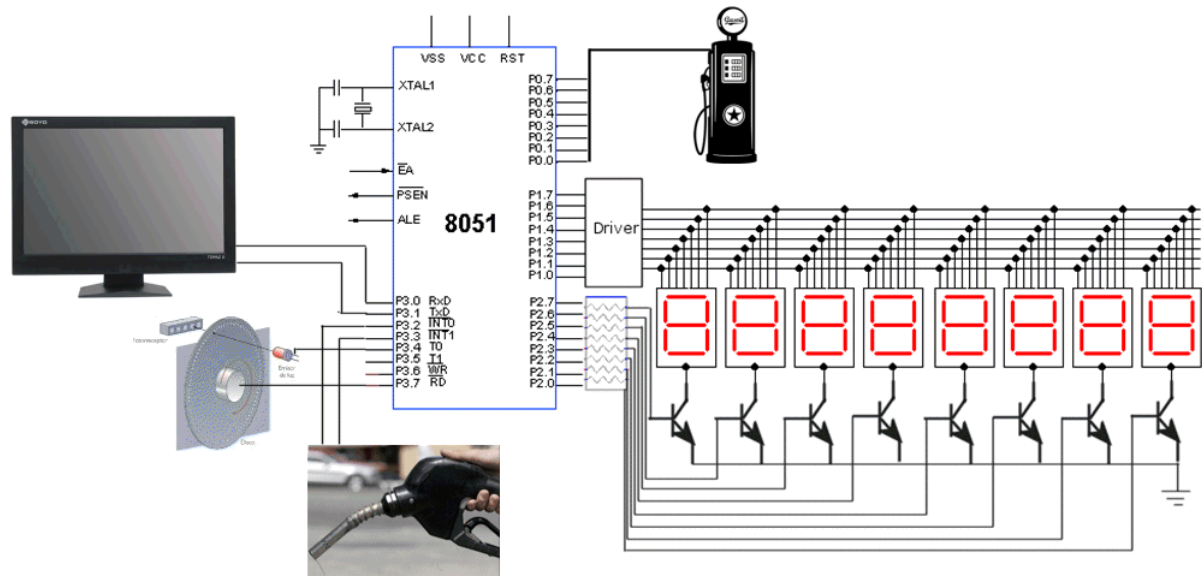
3) Conectar um teclado matricial conforme a figura, na Porta P1 do Microcontrolador.



Escrever um programa em Assembly que permita a um usuário digitar uma senha de 4 dígitos no teclado. Se a senha digitada for 4AD3 em hexadecimal, o programa deve acender o LED1 (P3.2) e preencher a RAM externa tal que o conteúdo de cada posição seja igual à parte menos significativa do endereço. Se a senha for CABE em hexadecimal a RAM externa deve ser totalmente zerada e o LED2 (P3.4) aceso. Qualquer outra senha que for digitada diferente das duas anteriores, o programa deve preencher a RAM externa com FFh, piscar os dois leds de maneira alternada (enquanto um está aceso o outro está apagado e vice-versa) em uma frequência aproximada de 10 Hz e não permitir a entrada de nenhuma outra senha.

Obs: Os Leds inicialmente devem ficar apagados.

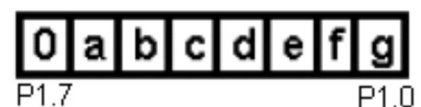
A figura abaixo é o circuito microprocessado de uma bomba de combustível e deverá ser usada para as questões 4 e 5.



Operação da bomba:

- O motor que bombeia o combustível é acionado pelo pino P3.7 = 1
- Cada litro que for bombeado equivale a um pulso na entrada T0
- Para ligar a bomba, o frentista aciona o gatilho que está ligado na Interrupção INT1, sensível a nível baixo (Baixa prioridade).
- Um sensor instalado no bico do alimentador de combustível está conectado na interrupção INT0, sensível à descida de borda, que detecta quando o tanque está cheio (Alta Prioridade).
- O display de 7 segmentos, catodo comum e multiplexado, mostra nos quatro dígitos menos significativos, acionados por P2.0 a P2.3, a quantidade de combustível transferida, e nos quatro dígitos mais significativos (P2.4 a P2.7) o valor em Reais (considerar 1 litro de combustível = 1 Real).
- Um terminal RS232 (9600, N, 8, 1) instalado no interior do posto de combustível é utilizado para a cobrança.
- A Porta P0 do microprocessador serve para a comunicação de dados com a bomba de combustível.

4) Considerando-se o display de 7 segmentos acionado pela Porta P1, escrever um Programa em Assembly que mostre nos Displays, a quantidade de combustível e o preço (em BCD) durante o tempo de enchimento do tanque. Observar que o motor da bomba é ligado (P3.7=1) quando o gatilho for acionado e desligado (P3.7=0) ou pelo sensor no bico do alimentador (INT0) ou pelo gatilho (INT1=1). Ao ser ligado o motor da bomba, os displays devem ser zerados e ao ser desligado os valores devem permanecer acesos no Display.



5) O preço a pagar e a quantidade de combustível bombeado devem ser transmitidos em ASCII para o terminal RS232 instalado no interior do posto no final de cada operação, ou seja, quando o motor da bomba for desligado. O Microprocessador desabilita as interrupções e aguarda um código de “transação OK” enviado pela linha serial (3Ch) do terminal, retransmitindo-o para a bomba de combustível através da Porta P0 e liberando a bomba para uma nova operação. A liberação da bomba deve ser realizada através da reabilitação das interrupções.

O cristal do microprocessador é de 11.0592 MHz.

Escrever este programa na forma de uma Sub-rotina e indicar em qual ponto no programa do item 1 ela deverá ser chamada.

Os esquemas baixo referem-se a um projeto com Microcontrolador MCS-51 e seus periféricos e deverá ser usado para responder às questões de 6 a 12.

A Tabela verdade de operação do decodificador SN74LS138 é:

DM74LS138

Inputs					Outputs							
Enable		Select										
G1	G2 (Note 1)	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	L	H	H	H	H	H	H	H
H	L	L	H	L	H	L	H	H	H	H	H	H
H	L	L	H	H	H	H	L	H	H	H	H	H
H	L	H	L	L	H	H	H	L	H	H	H	H
H	L	H	L	H	H	H	H	H	L	H	H	H
H	L	H	H	L	H	H	H	H	H	L	H	H
H	L	H	H	H	H	H	H	H	H	H	L	L

74HCT373 é um Latch de 8 Flip-flops tipo D

74HCT244 é um buffer de 8 portas não inversoras

ADC0808 é um conversor AD de 8 canais analógicos (mesmo dado em aula)

DAC0808 é um conversor DA de 8 Bits (mesmo de aula)

AM27C128 é uma memória

6) Preencher a tabela abaixo com o Mapeamento de Memória (endereço inicial e final) e com o nome do dispositivo mapeado para cada pino de seleção do decodificador (SN74LS138):

LINHA DE SELEÇÃO	ENDEREÇO INICIAL (Hexadecimal)	ENDEREÇO FINAL (Hexadecimal)	Dispositivo
Y0			
Y1			
Y2			
Y3			
Y4			
Y5			
Y6			
Y7			

7)Escrever um programa em Assembly do 8051 que seleciona o canal 7 do conversor AD, inicializa a conversão, lê o código gerado e o envia continuamente para o conversor DA.

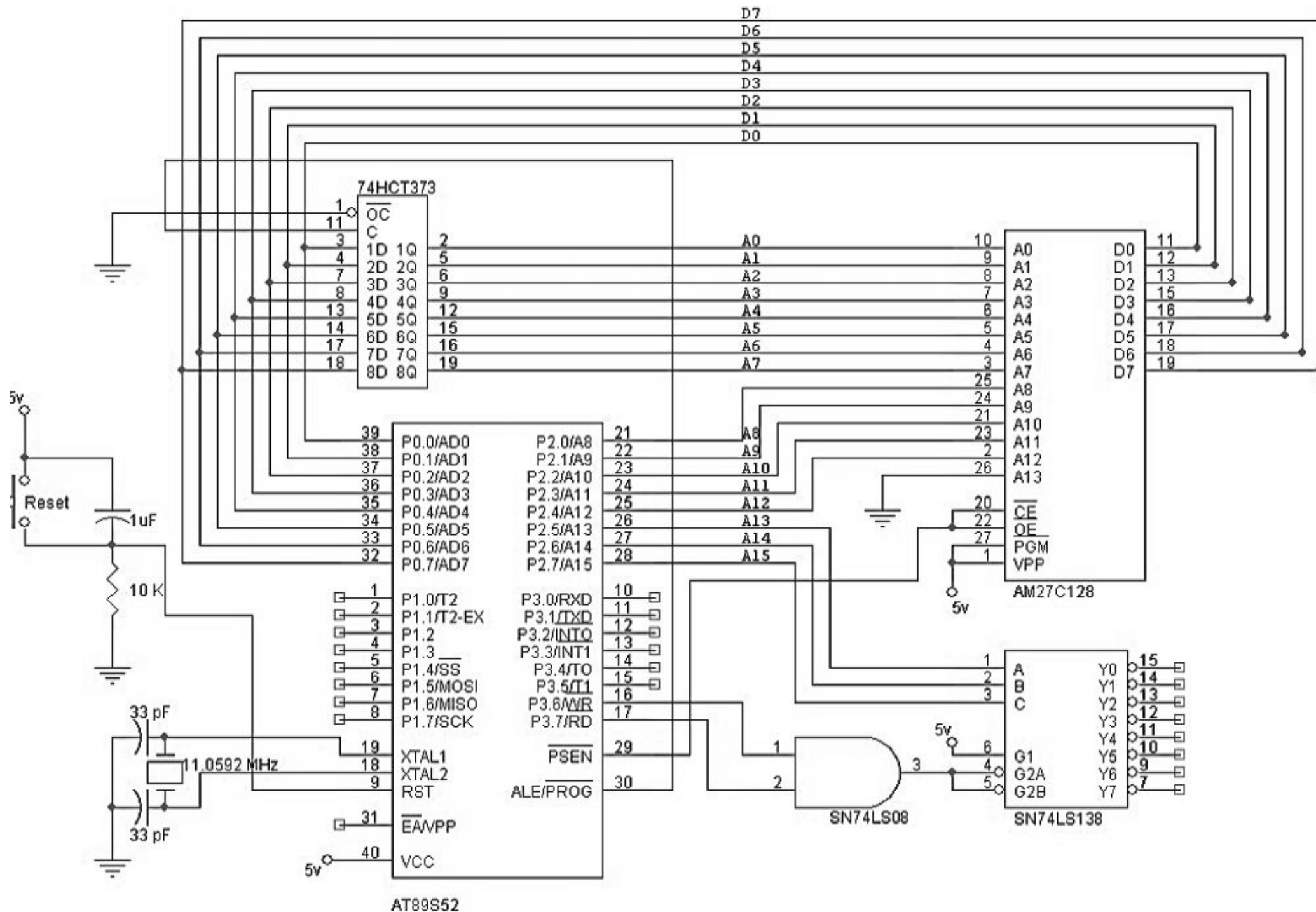
8)Escrever um programa em Assembly do 8051 que mostre um contador BCD crescente no display de 7 segmentos com intervalo de 1 segundo entre cada dígito mostrado.

9)Escrever um programa em Assembly do 8051 que leia o teclado matricial. Se o código inserido for ABCD o programa deve acender o dígito 1 no display de 7 segmentos. Se o código inserido for 0000 o programa deve acender o dígito 0 no display de 7 segmentos. Caso qualquer outro código diferente seja inserido, o display de 7 segmentos deve piscar com intervalo de 1 segundo, acendendo e apagando o dígito 8 em loop, não aceitando nenhum novo código.

10)Considerando que a cada volt na entrada analógica do conversor AD gera um código digital, ou seja, 0v = 00, 1v = 01, 2v = 02 e assim sucessivamente, desenvolver um programa em Assembly do 8051 que mostre o valor da tensão aplicada no display de 7 segmentos (0 a 9 volts).

11)Escrever um programa em Assembly do 8051 que ao se pressionar a tecla 1 no teclado matricial transfira o valor digital do canal 1 do conversor AD para o conversor DA e acenda o número do canal no display de 7 segmentos. O mesmo deve ocorrer para cada canal do conversor AD, ou seja: tecla 2 apertada no teclado matricial, transferir canal 2 do AD para o DA e acender 2 no display; tecla 3 apertada no teclado matricial, transferir canal 3 do AD para o DA e acender 3 no display; e assim sucessivamente até a tecla 8. O programa deve ficar em loop aceitando sempre qualquer das 8 teclas.

12)Conectar um monitor e teclado ASCII RS232 aos pinos P3.0 e P3.1 deste microcontrolador. Programar a comunicação serial assíncrona no padrão 19200,N,8,1. Fazer um programa em Assembly que aceite dados do teclado em ASCII para selecionar cada canal do conversor AD e escrever na tela do monitor o valor do código digital gerado em cada canal, ou seja, o dígito 1 em ASCII digitado no teclado seleciona o canal 1 do conversor AD e o valor hexadecimal convertido para ASCII é enviado para o monitor acompanhado da informação "CANAL 1 = hexa do canal 1". O mesmo deve ocorrer para cada canal do AD selecionado pelo teclado ASCII. O programa deve ficar em loop para aceitar teclas restritas entre 1 e 8 no teclado ASCII a qualquer momento.



Entradas Analógicas

